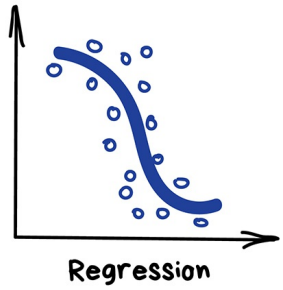
# Линейная Регрессия

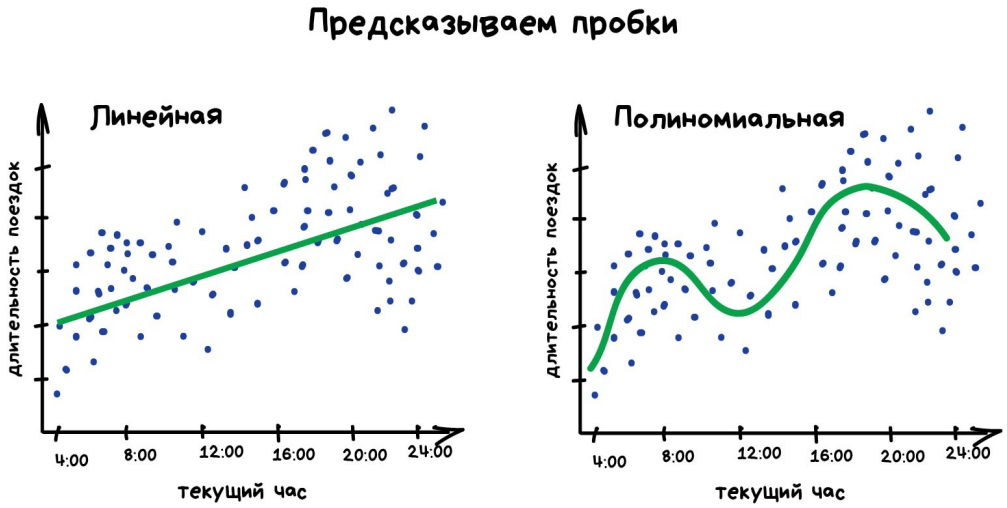



**Регре́ссия** (лат. *regressio* — обратное движение, отход) в теории вероятностей и математической статистике — односторонняя стохастическая зависимость, устанавливающая соответствие между случайными переменными.

Сегодня используют для:

- Прогноз стоимости ценных бумаг
- Анализ спроса, объема продаж
- Медицинские диагнозы
- Любые зависимости числа от времени

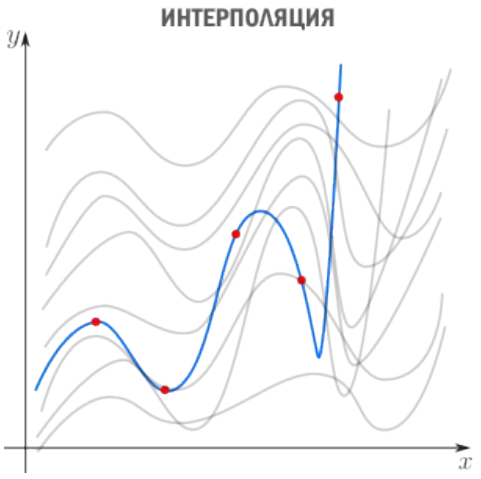
Регрессию очень любят финансисты и аналитики, она встроена даже в Excel. Внутри всё работает, опять же, банально: машина тупо пытается нарисовать линию, которая в среднем отражает зависимость.




Регрессия

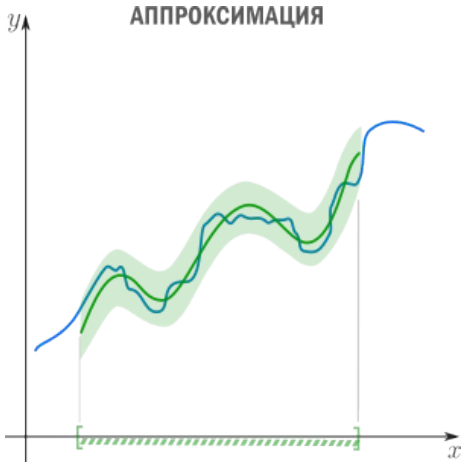
Когда регрессия рисует прямую линию, её называют линейной, когда кривую — полиномиальной. Это два основных вида регрессии. Но так как в семье не без урода, есть Логистическая Регрессия, которая на самом деле не регрессия, а метод классификации, от чего у всех постоянно путаница.

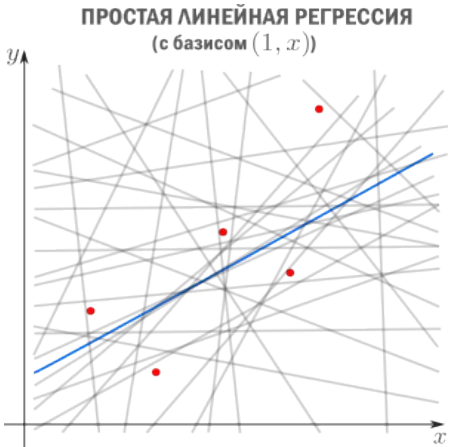
Есть три сходных между собой понятия, три сестры: интерполяция, аппроксимация и регрессия.У них общая цель: из семейства функций выбрать ту, которая обладает определенным свойством.




**Интерполяция** — способ выбрать из семейства функций ту, которая проходит через заданные точки. Часто функцию затем используют для вычисления в промежуточных точках. Например, мы вручную задаем цвет нескольким точкам и хотим чтобы цвета остальных точек образовали плавные переходы между заданными. Или задаем ключевые кадры анимации и хотим плавные переходы между ними. Классические примеры: интерполяция полиномами Лагранжа, сплайн-интерполяция, многомерная интерполяция (билинейная, трилинейная, методом ближайшего соседа и т.д). Есть также родственное понятие экстраполяции — предсказание поведения функции вне интервала. Например, предсказание курса доллара на основании предыдущих колебаний — экстраполяция.

**Аппроксимация** — способ выбрать из семейства «простых» функций приближение для «сложной» функции на отрезке, при этом ошибка не должна превышать определенного предела. Аппроксимацию используют, когда нужно получить функцию, похожую на данную, но более удобную для вычислений и манипуляций (дифференцирования, интегрирования и т.п). При оптимизации критических участков кода часто используют аппроксимацию: если значение функции вычисляется много раз в секунду и не нужна абсолютная точность, то можно обойтись более простым аппроксимантом с меньшей «ценой» вычисления. Классические примеры включают ряд Тейлора на отрезке, аппроксимацию ортогональными многочленами, аппроксимацию Паде, аппроксимацию синуса Бхаскара и т.п.




**Регрессия** — способ выбрать из семейства функций ту, которая минимизирует функцию потерь. Последняя характеризует насколько сильно пробная функция отклоняется от значений в заданных точках. Если точки получены в эксперименте, они неизбежно содержат ошибку

измерений, шум, поэтому разумнее требовать, чтобы функция передавала общую тенденцию, а не точно проходила через все точки. В каком-то смысле регрессия — это «интерполирующая аппроксимация»: мы хотим провести кривую как можно ближе к точкам и при этом сохранить ее максимально простой чтобы уловить общую тенденцию. За баланс между этими противоречивыми желаниями как-раз отвечает функция потерь (в английской литературе «loss function» или «cost function»).

Можно отметить схожесть регрессии и классификации, которая подтверждается еще и тем, что многие классификаторы, после небольшого тюнинга, превращаются в регрессоры. Например, мы можем не просто смотреть к какому классу принадлежит объект, а запоминать, насколько он близок — и вот, у нас регрессия.

## **Простая линейная регрессия** (пример)

Вообще множественный регрессионный анализ представляет собой целый класс методов в статистике. Однако базовый инструмент, лежащий в основе этих методов - один. Это *простая линейная регрессия*. В самой *простой линейной регрессии* нет ничего сложного, и, чтобы понять принцип её устройства, достаточно знать математику на уровне 10-11 класса. Так что попробуем разобраться, что это такое.

Лучше всего разбираться на примерах, поэтому представим себе следующую ситуацию. Мы собрали информацию о результатах ЕГЭ по математике для одного из выпускных классов, а также информацию о среднем количестве часов, потраченных школьниками на подготовку к экзамену в домашних условиях за месяц до экзамена. Эта информация была записана в следующем виде:

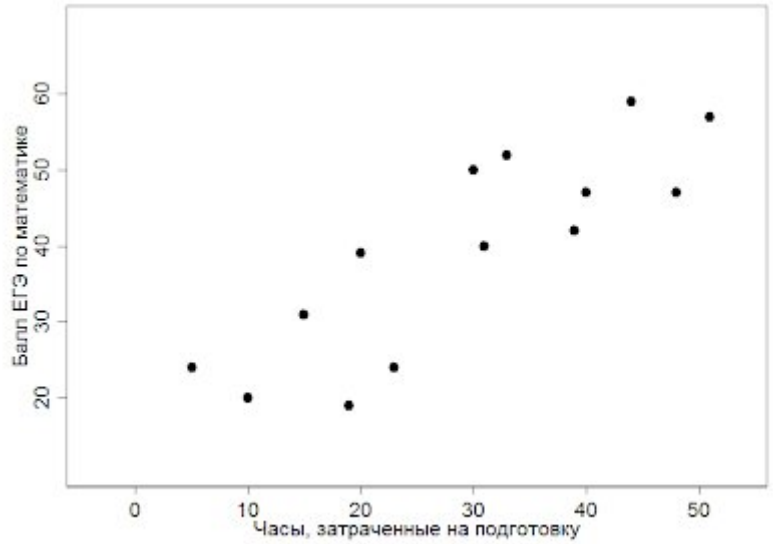
**Таблица 1**

| Номер учащегося | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Балл ЕГЭ по математике | 39 | 57 | 59 | 24 | 19 | 31 | 40 | 24 | 47 | 52 | 50 | 42 | 20 | 47 |
| Часы, затраченные | 20 | 51 | 44 | 23 | 19 | 15 | 31 | 5 | 40 | 33 | 30 | 39 | 10 | 48 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| на подготовку | | | | | | | | | | | | | | |

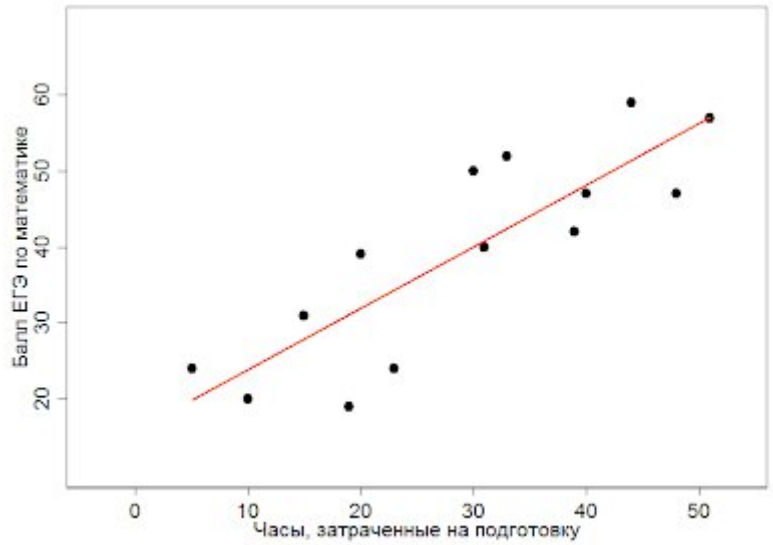
Если мы хотим продемонстрировать связь между баллами ЕГЭ и количестве часов, затраченных школьниками на подготовку, достаточно построить простой график (см. *рисунок 1*). Как правило, тот показатель, который мы хотим объяснить с помощью другого, располагается по оси *Y* (т.е. вертикальной оси). Объясняющий показатель располагается по оси *X* (т.е. горизонтальной оси):

**Рисунок 1**




Визуально не сложно убедиться, что связь есть: в среднем более высоким часам подготовки соответствует более высокий балл ЕГЭ. Кроме того, на *рисунке 1* можно мысленно провести линию между точками, которая с высокой вероятностью будет иметь положительной наклон - очень часто однозначная возможность провести такую в ходе визуального анализа уже сама по себе указывает на наличие сильной корреляции, т.е. связи между двумя показателями. На *рисунке 2* приведен пример такой линии (красным):

**Рисунок 2**



Так вот *простая линейная регрессия* - это способ проведения такой линии, только полученный не мысленно и не произвольно, а выводя ее строго математически, исходя из реального расположения точек на плоскости. Важно уяснить, что и линия, и регрессия являются моделью, т.е. некоторым (порой даже весьма серьезным!) упрощением реальности, воспроизводящим тем не менее некоторые существенные свойства этой реальности.

# Подготовка данных к линейной регрессии

Линейная регрессия изучается уже давно, и есть много литературы о том, как ваши данные должны быть структурированы, чтобы наилучшим образом использовать модель МНК или Градиентного спуска.

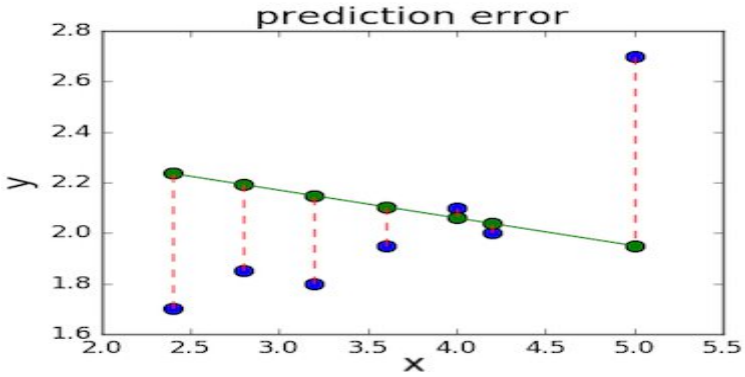
Таким образом, когда речь идет об этих требованиях и ожиданиях, они могут быть пугающими. Эти правила можно использовать скорее как практические правила при использовании алгоритмов линейной регрессии.

Используя эти эвристики и посмотреть, что лучше всего работает для вашей проблемы:

- **Линейные предпосылки**. Линейная регрессия предполагает, что связь между входными и выходным данными является линейной. Линейная регрессия не поддерживает ничего другого. Это может быть очевидно, но это хорошо, чтобы помнить, когда у вас есть много атрибутов. Возможно, потребуется преобразовать данные, чтобы сделать отношения между ними линейными (например, логарифмическое преобразование для экспоненциальной связи).
- **Удалите шум**. Линейная регрессия предполагает, что переменные на выходе и вывода не являются шумными. Рассмотрите возможность использования операций по очистке данных, которые позволяют лучше разоблачать и прояснять сигнал в данных. Это наиболее важно для переменной вывода, и, по возможности, необходимо удалить выбросы в переменной вывода (y).
- **Удалите коллинеарность.** Линейная регрессия будет чрезмерно соответствовать вашим данным, когда у вас есть сильно коррелированные входные переменные. Рассмотрим расчет парных корреляций для входных данных и удаление наиболее коррелированных данных.
- **Гауссово распределение**. Линейная регрессия сделает более надежные прогнозы, если входные и выходные переменные имеют гауссово распределение. Вы можете получить некоторую выгоду с помощью преобразований (например, log или BoxCox) на переменных, чтобы сделать их распределение более гауссово.
- **Нормализованные входные данные**: Линейная регрессия часто делает более надежные прогнозы, если отмасштабировать входные переменные с помощью стандартизации или нормализации.

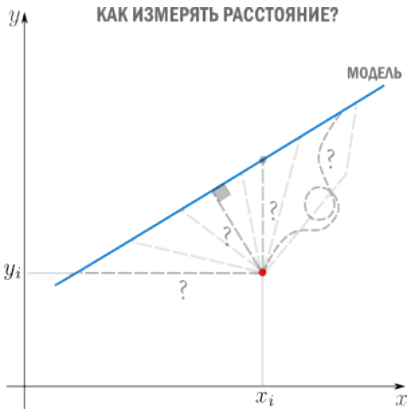
# Функция потерь — метод наименьших квадратов

Функция потерь — это мера количества ошибок, которые наша линейная регрессия делает на наборе данных. Хотя есть разные функции потерь, все они вычисляют расстояние между предсказанным значением $y(x)$ и его фактическим значением. Например, взяв строку из среднего примера выше, $f(x)=-0.11 \cdot x+2.5$, мы выделяем дистанцию ошибки между фактическими и прогнозируемыми значениями  красными пунктирными линиями.




# Метод наименьших квадратов

Начнём с простейшего двумерного случая. Пусть нам даны точки на плоскости и мы ищем такую аффинную функцию



чтобы ее график ближе всего находился к точкам. Таким образом, наш базис состоит из константной функции и линейной .

Как видно из иллюстрации, расстояние от точки до прямой можно понимать по-разному, например геометрически — это длина перпендикуляра. Однако в контексте нашей задачи нам нужно функциональное расстояние, а не геометрическое. Нас интересует разница между экспериментальным значением и предсказанием модели для каждого  поэтому измерять нужно вдоль оси .

Первое, что приходит в голову, в качестве функции потерь попробовать выражение, зависящее от абсолютных значений разниц .

Простейший вариант — сумма модулей отклонений  приводит к Least Absolute Distance (LAD) регрессии.

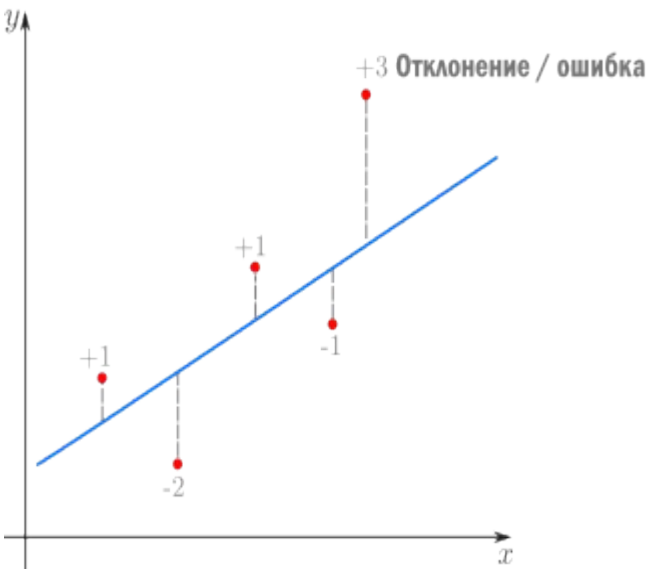
Впрочем, более популярная функция потерь — сумма квадратов отклонений регрессанта от модели. В англоязычной литературе она носит название Sum of Squared Errors (SSE)

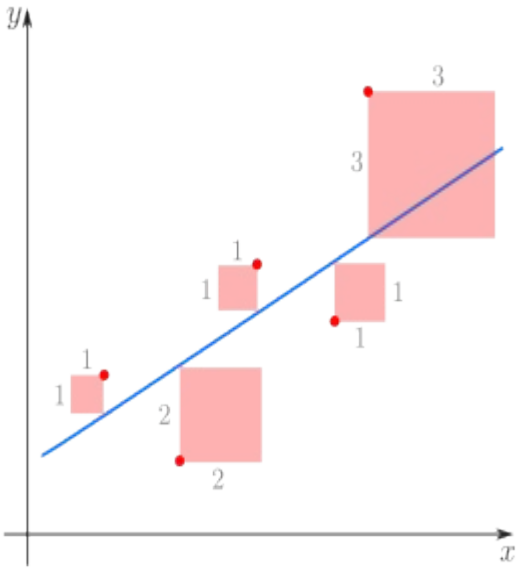
Если они имеют тенденцию располагаться по прямой, то следует искать **уравнение прямой** с оптимальными значениями  и .  Иными словами, задача состоит в нахождении ТАКИХ коэффициентов  – чтобы сумма квадратов отклонений

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

была наименьшей.

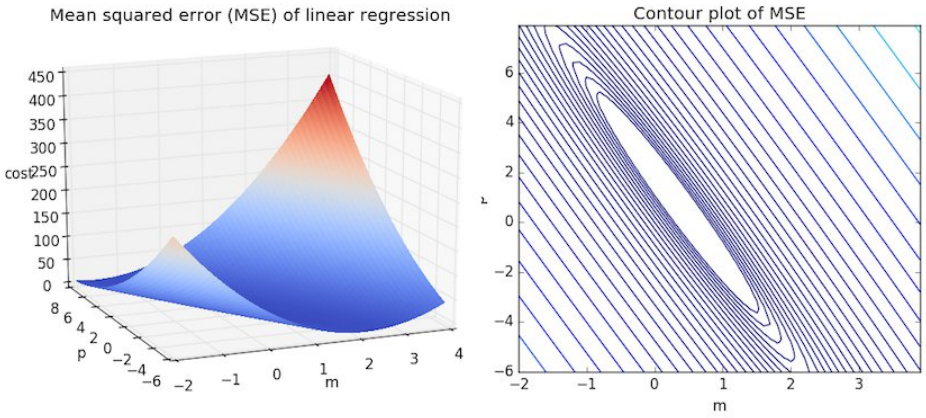
Метод наименьших квадратов (по англ. OLS) — линейная регрессия с  в качестве функции потерь.



Такой выбор прежде всего удобен: производная квадратичной функции — линейная функция, а линейные уравнения легко решаются.

Рассмотрим приведенный ниже рисунок, который использует две визуализации средней квадратичной ошибки в диапазоне, где наклон *m* находится между *-2* и *4*, а *b* между *-6* и *8*.




Слева: диаграмма, изображающая среднеквадратичную ошибку для $-2 \leq m \leq 4$, $-6 \leq p \leq 8$ Справа: тот же рисунок, но визуализирован как контурный график, где контурные линии являются логарифмически распределенными поперечными сечениями высоты.

Глядя на два графика, мы видим, что наш MSE имеет форму удлиненной чаши, которая, по-видимому, сглаживается в овале, грубо центрированном по окрестности $(m, p) \approx (0.5, 1.0)$. Если мы построим MSE линейной регрессии для другого датасета, то получим аналогичную форму. Поскольку мы пытаемся минимизировать MSE, наша цель — выяснить, где находится самая низкая точка в чаше.